# Założenia dla zadań projektowych

Zadania projektowe wykonywane są w dwóch grupach. Podział na grupy został przedstawiony w *załączniku 1*. Zadania projektowe należy realizować zgodnie z wybraną metodyką.

Produkt wynikowy z realizacji projektu musi składać się z:

* Kodu źródłowego aplikacji
  + Aplikacja musi zawierać elementy programowania zdarzeniowego
  + Aplikacja musi być zgodna i spójna z wybraną konwencją popularną dla danego języka programowania
  + Aplikacje muszą przechowywać dane w bazie danych
  + Aplikacja musi implementować 5 własnych zdarzeń, które w zależności od typu będą świadczyły o wykonaniu dedykowanej logiki
  + (?)Aplikacja musi implementować 3 komponenty graficzne własnego pomysłu
  + Aplikacja musi przechowywać dane w wybranej bazie danych, dostęp do baz danych powinien odbywać się z wykorzystaniem ORM lub korzystać z interfejsów baz obiektowych
  + Aplikacja musi być parametryzowana przez pliki properties/xml/yaml/json
  + (?)Aplikacja musi korzystać z jednego plikowego źródła danych (np słowniki)
  + Aplikacja musi podłączyć się do wybranej jednej usługi webowej i wyświetlić jej wynik w aplikacji (np. czas sieciowy, pogoda itp)
  + Aplikacja musi dostarczać mechanizmy logowania na wybranych poziomach szczegółowości
  + Należy zastosować wzorce projektowe programowania zdarzeniowego(SWING/Java FX)
  + Asynchroniczność wykonywanych operacji w tle (np. Swing Worker dla SWING, kolejki JMS,AJAX dla web)
  + Aplikacja musi dostarczać możliwość intercjonalizacji (zmiany języka) oraz zmiany wyglądu
  + Aplikacja musi korzystać z menadżera paczek (MAVEN, Nuget, Gradle, npm, pip)
  + Aplikacja napisana obiektowo zgodnie z paradygmatem SOLID
  + Aplikacja musi dostarczać mechanizmy autoryzacji i autentykacji, a także rozliczalności użytkowników

Sprawozdanie musi zawierać:

* opis projektu
* specyfikację przypadków użycia, aktorów, systemów zewnętrznych;
* specyfikację wymagań funkcjonalnych i niefunkcjonalnych;
* dla 3 głównych (nietrywialnych) przypadków użycia zamodelowane diagramy sekwencji dla ścieżek głównych i alternatywnych;
* logiczny model bazy danych/encji obietkowych;
* przygotowanie mockup’ów dla widoków w systemie (całość + 3 wybrane przypadki użycia)
* wylistowanie i krótkie opisanie wszystkich zastosowanych bibliotek;
* wylistowanie i przedstawienie wykorzystanych wszystkich wzorców projektowych;
* diagram klas/encji reprezentujący struktury bazodanowe wykorzystywane w projekcie;
* opisane wykorzystane metody autoryzacji i autentykacji;
* opracowanie perspektywy komponentów systemu;
* opracowanie perspektywy wdrożeniowej systemu;
* prezentacje wszystkich widoków aplikacyjnych ze stosowanymi opisami funkcjonalności i ról;
* rozmieszczenie wszelkich plików konfiguracyjnych wraz z opisem właściwości konfiguracyjnych wykorzystywanych w aplikacji;
* załączone kody źródłowe wszystkich aplikacji;
* opracowanie przypadków testowych dla 3 głównych przypadków użycia oraz ich weryfikacji
* szczegółową instrukcję użytkownika korzystania z aplikacji
* filmik prezentujący możliwości aplikacji (przedstawienie wszystkich widoków w aplikacji)
* instrukcję zbudowania i uruchomienia aplikacji, najlepiej konfigurację vagrant/docker do przygotowania środowiska developerskiego aplikacji

Kod powinien być przechowywany na platformie GitHub. W celu założenia repozytorium prywatnego należy uzyskać licencję Micro poprzez <https://education.github.com/pack/offers>.

Każda akcja wykonywana na repozytorium musi być opatrzona odpowiednim komentarzem.

# Plan rozliczania zadań

W poszczególnych terminach w udostępnionym katalogu na OneDrive należy udostępnić następujące produkty:

2.12.2018 - implementacja wybranych 4 wymagań implementacyjnych

23.12.2018 - implementacja wybranych, kolejnych 4 wymagań implementacyjnych

2.01.2019 – działająca wersja opracowanej aplikacji implementująca co najmniej 10 wymagań implementacyjnych, a także implementująca co najmniej 50% wszystkich przepływów wszystkich przypadków użycia

Koniec stycznia/ostatnie zajęcia – implementacja kolejnych wymagań – co najmniej 12 wymagane do zaliczenia, implementująca co najmniej 80% zamodelowanych przepływów przypadków użycia, opracowane sprawozdanie zgodnie z wymaganiami oraz gotowa aplikacja spełniająca wszystkie wymagania

Portal do prowadzenie rozgrywek internetowych (a la portal Kurnik) z implementacją jednej wybranej gry

3 komponenty graficzne własnego pomysłu

->Grafika gry

->Logo gry

->Logo strony

Autoryzacja i autentykacja -> do czego nam to jest potrzebne? Czy konto administratora?

Nazwa projektu bardziej uniwersalna

Opis projektu:

Celem projektu jest wykonanie portalu internetowego umożliwiającego granie w gry. Zalogowanym użytkownikom daje możliwość wybrania gry. Po zakończeniu rozgrywki zapisywany jest wynik danego użytkownika, a także wyświetla ranking najlepszych użytkowników.

Każdy użytkownik ma możliwość zmiany szaty graficznej i zmiany języka.

Wymagania funkcjonalne:

|  |  |  |
| --- | --- | --- |
| W\_01 | Zarejestrowanie | Wpisanie nazwy użytkownika oraz hasła. System sprawdza czy nazwa się nie powtarza, a w przypadku powodzenia dodaje użytkownika do bazy danych. |
| W\_02 | Zalogowanie | Wpisanie nazwy użytkownika oraz hasła. Jeśli wszystko się zgadza logowanie kończy się sukcesem, a w przeciwnym wypadku pojawia się komunikat i można ponownie spróbować się zalogować. |
| W\_03 | Sprawdzenie profilu | Wejście w odpowiednią zakładkę, w której wyświetlone są dane użytkownika – jego nazwa, ilość dni od rejestracji, najlepsze wyniki w danej grze wraz z datą. |
| W\_04 | Zmiana koloru strony | Zmiana motywu graficznego strony – tła i tekstu. |
| W\_05 | Zmiana języku strony | Zmiana języku strony na jeden z dostępnych do wyboru. |
| W\_06 | Wylogowanie | Wylogowanie zalogowanego użytkownika ze strony. |
| W\_07 | Wybranie gry | Wybranie jednej z dostępnych gier. |
| W\_08 | Granie | Przeprowadzenie rozgrywki w wybraną grę. |
| W\_09 | Wyświetlenie wyniku gry | Poza skończonej grze, wyświetlenie wyniku gracza i najlepszych użytkowników. |

Wymagania poza funkcjonalne:

|  |  |  |
| --- | --- | --- |
| W\_poza\_01 | Wydajny | Strona działa płynnie. |
| W\_poza\_02 | Czytelny | Interfejs użytkownika jest prosty i zrozumiały. Ikony przycisków są intuicyjne i nie ma poukrywanych funkcjonalności aplikacji. |
| W\_poza\_03 | Archiwizacja | Strona przekazuje do bazy danych informacje o wynikach rozgrywek. |
| W\_poza\_04 | Dostępny | Z systemu może korzystać każdy posiadający dostęp do internetu. |
| W\_poza\_05 | Niezawodny | Strona działa bez błędów. |

Specyfikacja przypadków użycia:

|  |  |
| --- | --- |
| Oznaczenie | PU1 |
| Nazwa | Zarejestruj |
| Priorytet | Wysoki |
| Opis | Założenie nowego konta użytkownika w portalu. |
| Aktorzy | Gracz |
| Warunki wstępne | Kliknięcie przycisku „zarejestruj” |
| Rezultat | Utworzenie nowego konta |
| Scenariusz | 1. Użytkownik klika przycisk rejestracji.  2. Użytkownik wypełnia formularz – podaje poprawny login oraz hasło.  3. Użytkownik zostaje dodany do bazy danych.  4. Nowe konto użytkownika zostaje dodane. |
| Scenariusz alternatywny | 1. Użytkownik klika przycisk rejestracji.  2. Użytkownik wypełnia formularz – podaje istniejący już login, lub hasło niespełniające wymagań.  3. Zostaje wyświetlony komunikat o błędzie przy rejestracji. |

|  |  |
| --- | --- |
| Oznaczenie | PU2 |
| Nazwa | Zalogowanie |
| Priorytet | Wysoki |
| Opis | Logowanie na konkretne konto |
| Aktorzy | Użytkownik |
| Warunki wstępne | Kliknięcie przycisku „zaloguj” |
| Rezultat | Zalogowanie w portalu |
| Scenariusz | 1. Użytkownik klika przycisk logowania.  2. Użytkownik wypełnia formularz – podaje poprawny login oraz hasło.  3. Serwer łączy się z bazą danych potwierdzając poprawność danych logowania.  4. Użytkownik zostaje zalogowany – zostaje wyświetlony komunikat o zalogowaniu. |
| Scenariusz alternatywny | 1. Użytkownik klika przycisk rejestracji.  2. Użytkownik wypełnia formularz – podaje niepoprawny login oraz hasło.  3. Serwer łączy się z bazą danych potwierdzając niepoprawność danych logowania.  4. Użytkownik nie zostaje zalogowany – zostaje wyświetlony komunikat o błędzie logowania. |

|  |  |
| --- | --- |
| Oznaczenie | PU3 |
| Nazwa | Sprawdzenie profilu |
| Priorytet | Umiarkowany |
| Opis | Wyświetlanie danych użytkownika – jego nazwa, ilość dni od rejestracji, najlepsze wyniki w danej grze wraz z datą. |
| Aktorzy | Użytkownik |
| Warunki wstępne | Kliknięcie przycisku „Profil”. |
| Rezultat | Wyświetla się strona z danymi użytkownika. |
| Scenariusz | 1. Użytkownik klika przycisk „Profil”.  2. Serwer łączy się z bazą danych w celu pobrania informacji na temat użytkownika.  3. Informacje zostają wyświetlone na odpowiedniej stronie. |

|  |  |
| --- | --- |
| Oznaczenie | PU4 |
| Nazwa | Zmiana koloru strony |
| Priorytet | Niski |
| Opis | Modyfikacja kolorystyki portalu |
| Aktorzy | Użytkownik |
| Warunki wstępne | Kliknięcie ikony odpowiedzialnej za zmianę kolorystyki |
| Rezultat | Konwersja kolorów na portalu. |
| Scenariusz | 1. Użytkownik klika ikonę odpowiedzialną za zmianę koloru.  2. Kolorystyka strony zmienia się. |

|  |  |
| --- | --- |
| Oznaczenie | PU5 |
| Nazwa | Zmiana języka strony |
| Priorytet | Niski |
| Opis | Strona zostanie wyświetlona w innym języku. |
| Aktorzy | Użytkownik |
| Warunki wstępne | Kliknięcie ikony odpowiedzialnej za zmianę języka. |
| Rezultat | Zmiana języka strony |
| Scenariusz | 1. Użytkownik klika ikonę odpowiedzialną za zmianę języka.  2. Język strony zmienia się. |

|  |  |
| --- | --- |
| Oznaczenie | PU6 |
| Nazwa | Wylogowanie |
| Priorytet | Umiarkowany |
| Opis | Wylogowanie użytkownika z portalu. |
| Aktorzy | Użytkownik |
| Warunki wstępne | Kliknięcie przycisku „Wyloguj” |
| Rezultat | Wylogowanie zalogowanego użytkownika. |
| Scenariusz | 1. Użytkownik klika ikonę „Wyloguj”  2. Użytkownik zostaje wylogowany. |

|  |  |
| --- | --- |
| Oznaczenie | PU7 |
| Nazwa | Wybranie gry |
| Priorytet | Umiarkowany |
| Opis | Użytkownik wybiera jedną z dostępnych gier, która zostanie uruchomiona. |
| Aktorzy | Użytkownik |
| Warunki wstępne | Kliknięcie ikony „Graj” przy wybranej grze. |
| Rezultat | Przygotowanie gry do uruchomienia. |
| Scenariusz | 1. Użytkownik klika przycisk „Graj”.  2. Zostaje wczytana strona danej gry, gotowej do uruchomienia. |

|  |  |
| --- | --- |
| Oznaczenie | PU8 |
| Nazwa | Granie |
| Priorytet | Bardzo wysoki |
| Opis | Przeprowadzenie rozgrywki w danej grze. |
| Aktorzy | Użytkownik |
| Warunki wstępne | Uruchomienie gry w menu gry |
| Rezultat | Rozgrywka |
| Scenariusz | 1. Użytkownik wybiera rozpoczęcie gry w menu gry.  2. Rozgrywka do czasu zakończenia gry. |

|  |  |
| --- | --- |
| Oznaczenie | PU9 |
| Nazwa | Wyświetlenie wyniku gry |
| Priorytet | Umiarkowany |
| Opis | Wyświetlenie wyniku gracza i najlepszych użytkowników. |
| Aktorzy | Użytkownik |
| Warunki wstępne | Zakończenie gry. |
| Rezultat | Wyświetlenie wyników. |
| Scenariusz | 1.Po zakończonej grze wyświetlają się rezultaty z rozgrywki.  2. Wybór następnej czynności. |









